Active Visual Segmentation

Ajay K. Mishra, Yiannis Aloimonos, Loong-Fah Cheong, and Ashraf A. Kassim, *Member, IEEE*

Abstract—Attention is an integral part of the human visual system and has been widely studied in the visual attention literature. The human eyes fixate at important locations in the scene, and every fixation point lies inside a particular region of arbitrary shape and size, which can either be an entire object or a part of it. Using that fixation point as an identification marker on the object, we propose a method to segment the object of interest by finding the "optimal" closed contour around the fixation point in the polar space, avoiding the perennial problem of scale in the Cartesian space. The proposed segmentation process is carried out in two separate steps: First, all visual cues are combined to generate the probabilistic boundary edge map of the scene; second, in this edge map, the "optimal" closed contour around a given fixation point is found. Having two separate steps also makes it possible to establish a simple feedback between the mid-level cue (regions) and the low-level visual cues (edges). In fact, we propose a segmentation refinement process based on such a feedback process. Finally, our experiments show the promise of the proposed method as an automatic segmentation framework for a general purpose visual system.

Index Terms—Fixation-based segmentation, object segmentation, polar space, cue integration, scale invariance, visual attention.

1 Introduction

The human (primate) visual system observes and makes sense of a dynamic scene (video) or a static scene (image) by making a series of fixations at various salient locations in the scene. The eye movement between consecutive fixations is called a saccade. Even during a fixation, the human eye is continuously moving. Such movement is called fixational movement. The main distinction between the fixational eye movements during a fixation and saccades between fixations is that the former is an involuntary movement whereas the latter is a voluntary movement [27]. But the important question is: Why does the human visual system make these eye movements?

One obvious role of fixations—the voluntary eye movements—is capturing high resolution visual information from the salient locations in the scene as the structure of the human retina has a high concentration of cones (with fine resolution) in the central fovea [38], [46]. However, psychophysics suggests a more critical role of fixations in visual perception. For instance, during a change blindness experiment, the subjects were found to be unable to notice a change when their eyes were fixated at a location away from where the change had occurred in the scene unless the change altered the gist or the meaning of the scene [19], [18]. In contrast, the change is detected quickly when the subjects fixated on the changing stimulus or close to it. This clearly suggests a more fundamental role of fixation in how we perceive a scene (or image).

The role of fixational eye movements—the involuntary eye movements—during a fixation is even more unclear. In fact, for a long time, these eye movements were believed to be just a neural tick and not useful for visual perception [22]. However, neuroscientists have recently revived the debate about the nature of these movements and their effects on visual perception [27], [16].

While we do not claim to know the exact purpose of these eye movements, we certainly draw our inspiration from the need of the human visual system to fixate at different locations in order to perceive that part of the scene. We think that fixation should be an essential component of any developed visual system. We hypothesize that, during a fixation, a visual system at least segments the region it is currently fixating at in the scene (or image). We also argue that incorporating fixation into segmentation makes it well defined.

1.1 Fixation-Based Segmentation: A Well-Posed Problem

In computer vision literature, segmentation essentially means breaking a scene into nonoverlapping, compact regions where each region constitutes pixels that are bound together on the basis of some similarity or dissimilarity measure. Over the years, many different algorithms [43], [35], [14] have been proposed that segment an image into regions, but the definition of what is a correct or "desired" segmentation of an image (or scene) has largely been elusive to the computer vision community. In fact, in our view, the current problem definition is not well posed.

To illustrate this point further, let us take an example of a scene (or image) shown in Fig. 1. In this scene, consider two of the prominent objects: the tiny horse and the pair of trees. Figs. 1b and 1c are the segmentation of the image using the normalized cut algorithm [35] for different input parameters (these outputs would also be typical of many other segmentation algorithms). Now, if we ask the question:

- *A.K. Mishra and Y. Aloimonos are with the Computer Vision Laboratory, Department of Computer Science, University of Maryland, A.V. Williams Bldg., College Park, MD 20742.*
E-mail: mishraka@umiacs.umd.edu, yiannis@cs.umd.edu.
- *L.-F. Cheong and A.A. Kassim are with the Electrical and Computer Engineering, National University of Singapore, Singapore 117576.*
E-mail: {eleclf, ashraf}@nus.edu.sg.

Manuscript received 15 June 2010; revised 4 June 2011; accepted 25 July 2011; published online 22 Sept. 2011.
Recommended for acceptance by Y. Ma.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-06-0450.
Digital Object Identifier no. 10.1109/TPAMI.2011.171.

0162-8828/12/$31.00 © 2012 IEEE Published by the IEEE Computer Society

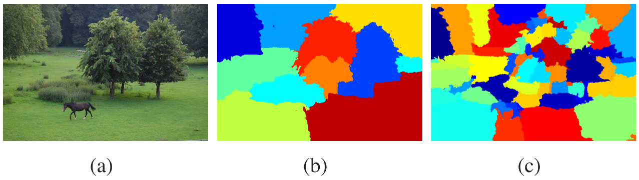


Fig. 1. Segmentation of a natural scene in (a) using the Normalized Cut algorithm [36] for two different values of its input parameter (the expected number of regions) 10 and 60 are shown in (b) and (c), respectively.

Which one of the two is the correct segmentation of the image? The answer to this question depends entirely on another question: What is the object of interest in the scene? In fact, there cannot be a single correct segmentation of an image unless it has only one object in prominence, in which case the correct segmentation of the image is essentially the correct segmentation of that object.

With respect to a particular object of interest, the correct/desired segmentation of the scene is the one wherein the object of interest is represented by a single or just a couple of regions. So, if the tiny horse is of interest, the segmentation shown in Fig. 1c is correct, whereas the segmentation shown in Fig. 1b is correct if the trees are of interest. Note, in Fig. 1b, the horse does not even appear in the segmentation. So, the goal of segmenting a scene is intricately linked with the object of interest in the scene and can be well defined only if the object of interest is identified and known to the segmentation algorithm beforehand.

But having to know about the object of interest even before segmenting the scene seems to make the problem one of many chicken-egg problems in computer vision, as we usually need to segment the scene first to recognize the objects in it. So, how can we identify an object even before segmenting it? What if the identification of the object of interest is just a weak identification such as a point on that object? Obtaining such points without doing any segmentation is not a difficult problem. It can be done using the visual attention systems, which can predict the locations in the scene that attracts attention [30], [41], [45], [10].

The human visual system has two types of attention: overt attention (eye movements) and covert attention (without eye movement). In this work, we mean overt attention whenever we use the term attention. The attention causes the eye to move and fixate at a new location in the scene. Each fixation will lie on an object, identifying that object (which can be a region in the background too) for the segmentation step. Now, segmenting that fixated region is defined as finding the "optimal" enclosing contour—a connected set of boundary edge fragments—around the fixation. This new formulation of segmenting fixated regions is a well-defined problem.

Note that we are addressing an easier problem than the general problem of segmentation where one attempts to find all segments at once. In the general segmentation formulation, the exact number of regions is not known and thus several ad hoc techniques have been proposed to estimate this number automatically. In fact, for a scene with prominent objects appearing at significantly different scales, having a single global parameter for segmenting the scene is not even meaningful, as explained above.

1.2 Overview

We propose a segmentation framework that takes as its input a fixation (a point location) in the scene and outputs the region containing that fixation. The fixated region is segmented in terms of the area enclosed by the "optimal" closed boundary around the fixation using the probabilistic boundary edge map of the scene (or image). The probabilistic boundary edge map, which is generated by using all available visual cues, contains the probability of an edge pixel being at an object (or depth) boundary. The separation of the cue handling from the actual segmentation step is an important contribution of our work because it makes segmentation of a region independent of types of visual cues that are used to generate the probabilistic boundary edge map.

The proposed segmentation framework is a two step process: First, the probabilistic boundary edge map of the image is generated using all available low-level cues (Section 3.2); second, the probabilistic edge map is transformed into the polar space with the fixation as the pole (Section 3.3), and the path through this polar probabilistic edge map (the green line in Fig. 6c) that "optimally" splits the map into two parts is found. This path maps back to a closed contour around the fixation point. The pixels on the left side of the path in the polar space correspond to the inside of the region enclosed by the contour in the Cartesian space, and those on the right side correspond to the outside of that region. Finding the optimal path in the polar probabilistic edge map is a binary labeling problem, and graph cut is used to find the globally optimal solution to this binary problem (Section 3.4).

1.3 Contributions

The main contributions of this paper are:

- Proposing an automatic method to segment an object (or region) given a fixation on that object (or region) in the scene/image. Segmenting the region containing a given fixation point is a well-defined binary labeling problem in the polar space, generated by transforming the probabilistic edge map from the Cartesian to the polar space with fixation point as pole. In the transformed polar space, the lengths of the possible closed contours around the fixation points are normalized (Section 3.1); thus, the segmentation results are not affected by the scale of the fixated region. The proposed framework does not depend upon any user input to output the optimal segmentation of the fixated region.
- Since we carry out segmentation in two separate steps, it provides an easy way to incorporate feedback from the current segmentation output to influence the segmentation result for the next fixation by just changing the probabilities of the edge pixels in the edge map. See Section 5 for how it is used in a multifixation framework to refine the segmentation output. Also, the noisy motion and stereo cues do not affect the quality of the boundary as the static monocular edges provide better localization of the region boundaries and the motion and stereo cues only help pick the optimal one for a given fixation.

2 Related Work

Although fixation is known be an important component of the human visual system, it has largely been ignored by computer vision researchers [32]. The researchers from visual attention, however, have investigated the reasons for the human visual system to fixate at certain salient points in the scene. The primary goal of such research has been to study the characteristics (e.g., color, texture) of the fixated location by tracking the eye of human subjects looking at still images or videos and use that information to make a prediction model that can estimate the possible fixation locations in the scene [30], [36], [41], [20], [45], [10]. The map showing the likelihood of each point in the image to be fixated by a human visual system is called saliency map. In [1], the saliency map is used to group the oversegmented regions, obtained using the mean-shift algorithm, into a bigger region representing the object. In essence, instead of using color information directly, they use the derived feature (saliency) to group the pixels together. So, it is in the spirit of any intensity or color-based grouping algorithm as per the segmentation step of the algorithm is concerned.

While visual attention research has made significant progress in making better predictions of what draws our attention [20], [31], [10], it does not explain what happens while the human visual system is fixated at a particular location in the scene. The human visual system spends significant amount of time fixated compared with the amount of time spent making saccades [17]. So, it is intuitive to think that the visual processing in the cortex is critically dependent on fixation. We propose a segmentation approach that takes the fixation as input and outputs a region. That way, any visual system can make a series of fixations at salient locations and perceives the scene in terms of the regions corresponding to these fixations.

There is a huge literature on various methods to segment images and videos into regions. Most segmentation algorithms depend upon some form of user input, without which the definition of the optimal segmentation of an image is ambiguous. There are two broad categories: first, the algorithms [35], [14], [43] that need various user-specified global parameters such as the number of regions and thresholds to stop the clustering; second, the interactive segmentation algorithms [6], [39], [4], [33] that always segment the entire image into only two regions: foreground and background. There are some hierarchical approaches [2] that do not require user input and they work well, especially for the image with a single object in prominence. Martínez et al. [26] correctly identifies the problems with the Normalized-Cut-based method [35] and proposes a solution to automatically select the global parameter for the segmentation process. But, since the cost of a cut is still computed in the Cartesian space, the "short-cut problem," explained later in Section 3.1, might still be an issue.

Boykov and Jolly [6] pose the problem of foreground/background segmentation as a binary labeling problem which is solved exactly using the max-flow algorithm [7]. It requires users to label some pixels as foreground or background to build their color models. Blake et al. [5] improved upon Boykov and Jolly [6] by using a Gaussian mixture Markov random field to better learn the foreground and background models. Rother et al. [33] requires users to specify a bounding box containing the foreground object. Arbelaez and Cohen [3] require a seed point for every region in the image. For foreground/background segmentation, at least two seed points are needed. These approaches report impressive results given appropriate user inputs. Stella and Shi [39] automatically selects multiple seed points by using spatial attention-based methods and then use these seed points to introduce extra constraints into their normalized cut-based formulation.

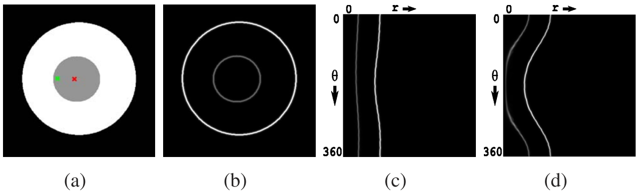
Unlike the interactive segmentation methods mentioned above, [44], [4] need only a single seed point from the user. Veksler [44] imposes a constraint on the shape of the object to be a star, meaning the algorithm prefers to segment convex objects. Also, the user input for this algorithm is critical as it requires the user to specify the center of the star shape exactly in the image. Bagon et al. [4] need only one seed point to be specified on the region of interest and segments the foreground region using a compositional framework. The algorithm outputs multiple disconnected regions as foreground even when the input seed point lies inside only one of those regions. It is computationally intensive and merges oversegmented regions, as is the case for many segmentation approaches [42], to form the final segmentation. It means that the mistakes made in the oversegmentation stage cannot be corrected. In contrast to this, our method makes only one decision about the region boundary and that is at the end of processing. In addition, the so-called "seed" point, in our case, is meaningful and is motivated from the fixation point of the human visual system.

Kolmogorov et al. [21] combine color, texture, and stereo cues to segment a binocular video into foreground and background regions. The computation of disparity values occurs simultaneously with the segmentation of the foreground. The video, however, should be captured with static cameras. In this paper, we segment the videos captured with a moving camera and with multiple independently moving objects in the scene. Also, we compute the low-level cues like color, texture, stereo, and motion separately and use all the cues only to create a better probabilistic boundary map. The segmentation step of finding the optimal closed boundary is only affected by the probabilistic boundary map.

3 Segmenting Fixated Region

As stated earlier in Section 1.2, segmenting a fixated region is equivalent to finding the "optimal" closed contour around the fixation point. This closed contour should be a connected set of boundary edge pixels (or fragments) in the edge map. However, the edge map contains both types of edges, namely boundary (or depth) and internal (or texture/intensity) edges. In order to trace the boundary edge fragments in the edge map to form the closed contour enclosing the fixation point, it is important to be able to differentiate between the boundary edges from the non-boundary (e.g., texture and internal) edges.

We generate a probabilistic boundary edge map of the scene wherein the intensity of an edge pixel is proportional to its probability to be at an object (or depth) boundary. The intensity ranges from 0 to 1. In qualitative terms, the boundary edges will appear brighter (darker) than the



(a) (b) (c) (d)

Fig. 2. (c) and (d) are the polar gradient edge maps generated by transforming the gradient edge map of the disc with respect to the fixations in red and green), respectively.

internal and texture edges in the (inverse) probabilistic boundary edge map. All available visual cues are used to generate such an edge map. The static cues (e.g., color and texture) are used to generate an initial edge map which is modified using stereo or motion cues. The detailed discussion on how we use binocular cues along with static cues to generate the probabilistic boundary edge map is given in Section 3.2.

Any algorithm that traces the closed contour through the probabilistic boundary edge map in the Cartesian space inherently prefers smaller contours as the overall cost, which essentially is the product of the length of the closed contour and the average cost of tracing the edge pixel along the contour, increases with their size. For possible closed contours with similar average boundary probabilities for their edge pixels, the scale makes smaller contours preferable to bigger contours. Our solution to the scale problem is to transform the edge map from the Cartesian to the polar Coordinate system (Section 3.1) and to segment the polar probabilistic boundary edge map to find the closed contour (see Section 3.4).

3.1 Polar Space Is the Key!

Let us consider finding the optimal contour around the red fixation point on the disc shown in Fig. 2a. The gradient edge map of the disc, shown in Fig. 2b, has two concentric circles. The big circle is the actual boundary of the disc, whereas the small circle is just the internal edge on the disc. The edge map correctly assigns the boundary contour intensity as 0.78 and the internal contour 0.39 (the intensity values range from 0 to 1). The lengths of the two circles are 400 and 100 pixels. Thus, the cost of tracing the boundary and the internal contour in the Cartesian space will be, respectively, $88 = (400 \times (1 - 0.78))$ and $61 = (100 \times (1 - 0.39))$. Clearly, the internal contour costs less, and hence it will be considered optimal even though the boundary contour is the brightest and should actually be the optimal contour. In fact, this problem of inherently preferring short contours over long contours has already been identified in the graph cut-based approaches where the minimum cut usually prefers to take a "short cut" in the image [37].

To fix this "short cut" problem, we have to transfer these contours to a space where their lengths no longer depend upon the area they enclose in the Cartesian space. The cost of tracing these contours in this space will then be independent of their scales in the Cartesian space. The polar space has this property, and we use it to solve the scale problem. The contours are transformed from the Cartesian coordinate system to the polar coordinate system with the red fixation point in Fig. 2b as the pole. In the polar space, both contours become open curves, spanning the entire θ axis, starting from 0 to 360 degrees. See Fig. 2c. Thus, the costs of tracing the inner contour and the outer contour become $80.3 = 365 \times (1 - 0.78)$ and $220.21 = 361 \times (1 - 0.39)$, respectively. As expected, the outer contour (the actual boundary contour) now costs the least in the polar space and hence becomes the optimal enclosing contour around the fixation.

It is important to make sure that the optimal path in the polar space is stable with respect to the location of the fixation, meaning that as the fixation point moves to a new location, the optimal path in the polar space for this new fixation location should still correspond to the same closed contour in the Cartesian space. For the new fixation point (the green "X") in Fig. 2b, both contours have changed shape (see Fig. 2d), but the "optimal" (or brightest) contour remains the same. Detailed discussion on the issue of stability with respect to the change in fixation location is done in Section 7.1.

3.2 Probabilistic Boundary Edge Map by Combining Cues

In this section, we carry out the first step of the segmentation process: generating the probabilistic boundary edge map using all available visual cues. There are two types of visual cues on the basis of how they are calculated: 1) static cues that come from just a single image; 2) stereo and motion cues that need more than one image to be computed. The static cues such as color, intensity, or texture can precisely locate the edges in the scene, but cannot distinguish between an internal texture edge from an edge at a depth discontinuity. On the other hand, stereo and motion can help distinguish between boundary and internal edge as there is a sharp gradient in disparity and flow across the former and no significant change across the latter. But unlike static cues, the stereo and motion cues are generally inaccurate at the boundary. This leads to the need to use the stereo and(or) motion cues and the static cues together such that they both identify and precisely locate the boundary edges in the scene.

3.2.1 Using Static Cues Only

Let us first consider the case when we only have a single image without any motion or stereo cues to help disambiguate the boundary edges from the rest. In that case, we need some intelligent way to make the distinction between edges. Let us start with the Canny edge map (Fig. 3b) of the image (Fig. 3a). The Canny edge detector finds edges at all the locations where there is a gradient in the intensity and returns a binary edge map, meaning all edge pixels are equally important. This makes the binary edge map useless for our purpose. However, if we assign the magnitude of the gradients at these locations as their respective probability of being at the boundaries, we have a meaningful boundary edge map. But it still has two main problems: First, the gradient magnitude is not always a good indicator of whether an edge pixel is at a boundary or not; second, Canny or similar intensity-based edge detectors are unable to find boundaries between textures and rather create strong edge responses inside a textured region.

Recently, an edge detector has been proposed by Martin et al. [24] that learns, using a linear classifier, the color and texture properties of the pixels across boundary edges

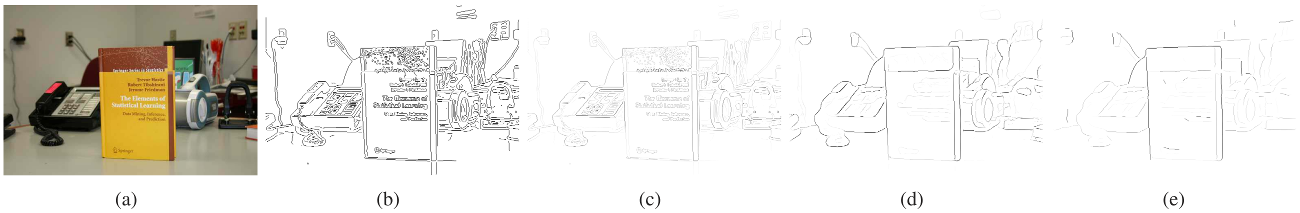

(a) (b) (c) (d) (e)

Fig. 3. Inverse probabilistic edge maps of the color image shown in (a). Darker pixels mean higher probability. (b) The Canny edge map. (c) The gradient edge map. (d) The output of the Berkeley pb detector [24]. (e) The final probabilistic boundary edge detector on combining static static cues with the motion cue.

versus that across the internal edges from a data set containing human-labeled segmentations of 200 images. The learned classifier is then used to assign appropriate probability (between 0 and 1) to the computed edges to be at a region boundary. Additionally, this edge detector handles texture in the image better than Canny or any intensity-based edge detectors. (See Fig. 3d. The spurious texture edges from Fig. 3c have been successfully removed.)

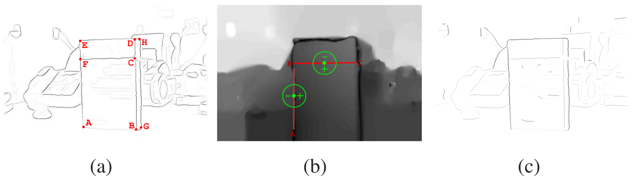
For single images, we are going to use the output of the Berkeley edge detector as the probabilistic boundary edge map to segment the fixation regions, which is explained later in the paper. Since this probabilistic edge map is calculated only out of color and texture cues, the edge map expectedly has strong internal edges (See BC, CD, CF in Fig. 4a) which are not actual depth boundaries.

To differentiate the internal edges from boundary (depth) edges, the stereo (and) or the motion cues are used. At a depth discontinuity or the boundary of a moving object, the optical flow and the disparity value change significantly. Also, inside an object, the disparity or the flow values remain largely unchanged. Based on this logic, the edge map is modified such that the edge pixels with strong gradient of either disparity or flow values across them are stronger (hence, have higher probability) than the ones with no gradient across them, which are essentially the internal edges on the objects.

3.2.2 Using Stereo and Static Cues

Let us combine stereo with static cues. We compute a dense disparity map for a pair of rectified stereo pair using the algorithm proposed by Ogale and Aloimonos [30]. Let us say, the range of disparity values lies between 0 and maximum value D. Our objective here is to use these disparity values to decide if an edge pixel is at a depth discontinuity.

Depth discontinuity causes a sudden change in the disparity values and the amount of change depends on the actual physical depth variation at the edge and the camera configuration. Also, the disparity values does not change across the internal edges on the object, barring small variations due to the error in the disparity map itself. So, the edge pixel with considerable change in the disparity values is considered to be a boundary edge. On the other hand, the edge pixels with a slight change in the disparity value are considered internal edges.



(a) (b) (c)

Fig. 4. Reinforcing the depth boundaries and suppressing the internal edges in the boundary edge map generated using static cues, shown in (a), to generate the final depth boundary, shown in (c), using the magnitude of the flow values, shown in (b).

Our approach to using relative disparity across the edge pixels to change their boundary probability is in agreement with the finding of the neurophysiological study [41] that the depth perception in a monkey brain depends upon the relative and not the absolute disparity. But how a given amount of relative depth change maps to the boundary probability is an important question which we cannot answer precisely. Logically speaking, the amount of change in the disparity should not matter as it occurs due to the relative placement of the objects in the scene. A depth boundary between two closely placed occluding objects should be as good a depth boundary as the one between a tree against a background far away from it.

To calculate the disparity change at an edge pixel, we place a circular disc with opposite polarity in the two halves separated by the diameter oriented along the tangent at the edge pixel (see Fig. 4b), and accumulate the disparity values from all pixels inside of the disc. The absolute value of the sum represents the difference in the average disparity on the both sides of the edge pixel. The radius of the disc is proportional to the image size. For our experiments, it is 0.02 of the image diagonal. Also, the disc template is weighted by its size to remove any affect of scale on this calculation. The reason to accumulate the change over a neighborhood around an edge pixel is to make sure that the presence of noise does not affect the process of finding depth edges.

Now that we have calculated the average change in disparity for an edge pixel, denoted by Δd, we have to map this to a probability value. To do that, we use a logistic function $P(\Delta d)$ given in (1). In this function, the ratio of the two parameters, β_2/β_1, represents the threshold over which the value of disparity change means the presence of a depth boundary. Also, there is a range around this threshold in which the probability changes from 0 to 1:

$$P(x) = \frac{1}{1 + exp(-\beta_1 x + \beta_2)}. \tag{1}$$

The parameters (β_1 and β_2) are learned using logistic regression on the two sets of depth gradients: one for the edge pixels on the boundary of objects, and the other for the edge pixels inside the objects. To select these edge pixels, five objects in a stereo pair are manually segmented. We collected the gradient values randomly at 200 boundary and internal edge locations. After logistic regression, the

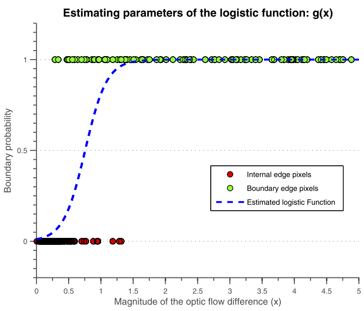


Fig. 5. The estimated logistic function converting optical flow gradient into probability is shown in blue.

parameters are found to be $\beta_1 = 2.4$ and $\beta_2 = 1.3$. (The measurements are in pixel units.)

3.2.3 Using Motion and Static Cues

Motion is different from stereo for two main reasons: First, unlike stereo, where a nonboundary edge does not have disparity change across it, an internal edge can also have a valid change in the flow across it. For instance, if a flat wheel is spinning along its axis, the flow vectors change direction across the spokes of the wheel, which are actually internal edges. Second, the optical flow vector representing motion information is a 2D vector, whereas the disparity is a scalar quantity making it easier to calculate their gradient than for the flow vector.

It is beyond the scope of this paper to define what constitutes a consistently moving object in terms of the flow vectors on them. Instead, we consider any change in the magnitude of the flow vector across an edge as a measure of discontinuity in depth across that edge. This definitely holds well when the relative motion between an object and the camera is translation in the X-Y plane only. As the motion in most videos primarily involves translational motion, the assumption holds good for all of them as it is evident in our experiments.

Just like the stereo case, we calculate the absolute change in the x-component and y-component of the optical flow map across an edge pixel separately using the oriented circular discs, and let us say ΔU and ΔV represent the changes, respectively. Then, the flow gradient across the edge pixel is given by $\sqrt{\Delta U^2 + \Delta V^2}$. Once again, the gradient value maps to a probability through the logistic function given in (1). Just like the stereo case, we train the parameters of the logistic function using the optical flow gradients from both at the boundary and inside of five moving objects in three different videos. The parameters of (1) are estimated to be $\beta_1 = 5.5$ and $\beta_2 = 4.2$. Fig. 5 shows the estimated logistic function as well as the training data.

An example of how the motion cue identifies the depth boundaries in the image is shown in Fig. 4, wherein the internal edge are clearly fainter (low probability) and the boundary edges are darker (high probability). With the improved boundary edge map, as the algorithm traces the brightest closed contour (AGHEA shown in Fig. 4a) around the fixation point, it will also be the real depth boundary of the region containing the fixation (Fig. 6d). In our experiments with videos, we have used the optical flow algorithm proposed by Brox et al. [9].

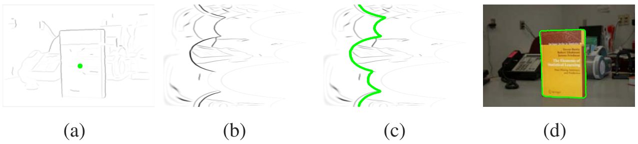

(a) (b) (c) (d)

Fig. 6. (a) The inverse probabilistic boundary edge map after combining motion cues with monocular cues. The fixation is shown by the green circular dot. (b) The polar edge map generated using the fixation as the pole. (c) The optimal contour through the polar edge map, splitting it into two parts: inside (left) and outside (right). (d) The closed contour around the fixation when transferred back to the Cartesian space.

Before proceeding to the next stage of finding the closed contour in the probabilistic boundary edge map around a given fixation, it is important to note that, in order for all points in the image to have a valid closed contour, the image borders have to be added as the edges. They will ensure enclosedness even for the fixations lying on the regions only partially present in the image. See, for instance, the car in column 5 of Fig. 15. A part of its closed boundary has to be the left border of the image. To make sure that they are preferred over the real edges, the intensity of the border edges corresponding is kept low.

3.3 Cartesian to Polar Edge Map

Let us say $E_{pb}^{pol}$is the corresponding polar plot of the probabilistic boundary edge map E_{pb} in the Cartesian space, and $F(x_o, y_o)$ is the selected pole (that is the fixation point). Now, a pixel $E_{pb}^{pol}(r, \theta)$ in the polar coordinate system corresponds to a subpixel location $\{(x, y) : x = rcos\theta + x_o, y = rsin\theta + y_o\}$ in the Cartesian coordinate system. $E_{pb}(x, y)$ is typically calculated by bilinear interpolation, which only considers four immediate neighbors.

We propose to generate a continuous 2D function $W(.)$ by placing 2D Gaussian kernel functions on every edge pixel. The major axis of these Gaussian kernel functions is aligned with the orientation of the edge pixel. The variance along the major axis is inversely proportional to the distance between the edge pixel and the pole O. Let S be the set of all edge pixels. The intensity at any subpixel location (x, y) in Cartesian coordinates is

$$W(x, y) = \sum_{i \in S} exp\left(-\frac{x_i^t}{\sigma_{x_i}^2} - \frac{y_i^t}{\sigma_{y_i}^2}\right) \times E_{pb}(x_i, y_i),$$

$$\begin{bmatrix} x_i^t \\ y_i^t \end{bmatrix} = \begin{bmatrix} cos\theta_i & sin\theta_i \\ -sin\theta_i & cos\theta_i \end{bmatrix} \begin{bmatrix} x_i - x \\ y_i - y \end{bmatrix},$$

where

$$\sigma_{x_i}^2 = \frac{K_1}{\sqrt{(x_i - x_o)^2 + (y_i - y_o)^2}}, \; \sigma_{y_i}^2 = K_2, \theta_i$$

is the orientation at the edge pixel i, $K_1 = 900$ and $K_2 = 4$ are constants. The reason for setting the square of variance along the major axis, $\sigma_{x_i}^2$, to be inversely proportional to the distance of the edge pixel from the pole is to keep the gray values of the edge pixels in the polar edge map the same as the corresponding edge pixel in the Cartesian edge map. The intuition behind using variable width kernel functions for

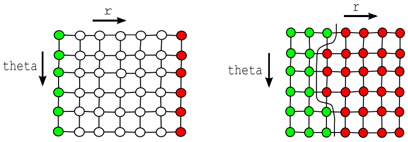


Fig. 7. Left: The green nodes in the first column are initialized to be inside, whereas the red nodes of the last column are initialized to be outside the region of interest. Right: The output of the binary labeling output after minimizing the energy function using graph cut. Note that though the first and the last rows in our graph are connected, they are not shown connected by an edge here for the sake of clarity.

different edge pixels is as follows: Imagine an edge pixel being a finite-sized elliptical bean aligned with its orientation and you look at it from the location chosen as the pole. The edge pixels closer to the pole (or center) will appear bigger and those farther away from the pole will appear smaller.

The polar edge map $E_{pb}^{pol}(r,\theta)$ is calculated by sampling $W(x,y)$. The values ofE_{pb}^{pol} are scaled to lie between 0 and 1. An example of such polar edge map is shown in Fig. 6b. Our convention is that the angle $\theta \in [0^\circ, 360^\circ]$ varies along the vertical axis of the graph and increases from the top to the bottom, whereas the radius $0 \leq r \leq r_{max}$ is represented along the horizontal axis increasing from left to the right. r_{max} is the maximum euclidean distance between any two locations in the image.

3.4 Finding the Optimal Cut through the Polar Edge Map: An Inside versus Outside Segmentation

Let us consider every pixel $p \in P$ of E_{pb}^{pol}as a node in a graph. Every node (or pixel) is connected to their four immediate neighbors (Fig. 7). A row in the graph represents the ray emanating from the fixation point at an angle (θ) equal to the row index. The first and the last rows of the graph are the rays $\theta = 0^\circ$and $\theta = 360^\circ$, respectively, which are essentially the same ray in the polar representation. Thus, the pairs of nodes $\{(0^\circ, r),\ (360^\circ, r)\},\ \forall r \in [0, r_{max}]$ should be connected by edges in the graph. The set of all the edges between neighboring nodes in the graph is denoted by Ω. Let us assume $l = \{0, 1\}$ are the two possible labels for each pixel where $l_p = 0$ indicates "inside" and $l_p = 1$ denotes "outside." The goal is to find a labeling $f(P) \mapsto l$ that corresponds to the minimum energy where the energy function is defined as

$$Q(f) = \sum_{p \in P} U_p(l_p) + \lambda \sum_{(p,q) \in \Omega} V_{p,q}.\delta(l_p, l_q), \tag{2}$$

$$V_{p,q} = \begin{cases} exp\big(-\eta E_{pb,pq}^{pol}\big) & \text{if } E_{pb,pq}^{pol} \neq 0, \\ k & otherwise, \end{cases} \tag{3}$$

$$\delta(l_p, l_q) = \begin{cases} 1 & \text{if } l_p \neq l_q, \\ 0 & otherwise, \end{cases} \tag{4}$$

where $\lambda = 50,\ \eta = 5,\ k = 20,\ E_{pb,pq}^{pol} = (E_{pb}^{pol}(r_p, \theta_p) + E_{pb}^{pol}(r_q, \theta_q))/2$, $U_p(l_p)$ is the cost of assigning a label l_p to the pixel p and $V_{p,q}$ is the cost of assigning different labels to the neighboring pixels p and q.

There is no information about how the color information of the inside and outside of the region containing the fixation. So, the data term U for all the nodes in the graph except those in the first column and the last column is zero: $U_p(l_p) = 0, \forall p \in (r, \theta), 0 < r < r_{max}, 0^\circ \leq \theta \leq 360^\circ$. However, the nodes in the first column that correspond to the fixation point in the Cartesian space must be inside and thus are initialized to the label 0: $U_p(l_p = 1) = D$ and $U_p(l_p = 0) = 0$ for $p \in (0, \theta), 0^\circ \leq \theta \leq 360^\circ$. The nodes in the last column, on the other hand, must lie outside the region and are initialized to the label 1: $U_p(l_p = 0) = D$ and $U_p(l_p = 1) = 0$ for $p \in (r_{max}, \theta), 0^\circ \leq \theta \leq 360^\circ$. See Fig. 7. In our experiments, we choose D to be 1,000; the high value is in order to make sure the initial labels to the first and the last columns do not change as a result of minimization. We use the graph cut algorithm [8] to minimize the energy function, $Q(f)$. The binary segmentation step splits the polar edge map into two parts: left side (inside) and right side (outside). The binary segmentation is finally transferred back to the Cartesian space to get the desired segmentation. For example, see Fig. 6c and Fig. 6d.

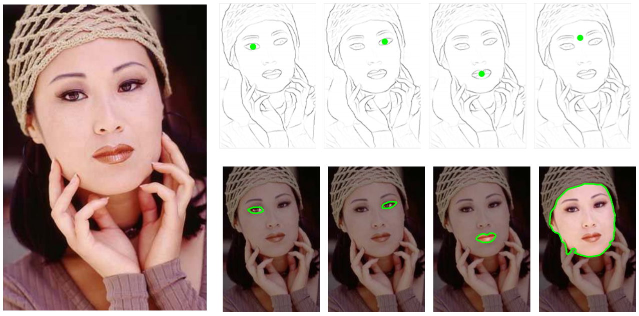


Fig. 8. The fixations, indicated by the green circular dots on the different parts of the face, are shown overlaid on the inverse probabilistic edge map of the leftmost image. The segmentation corresponding to every fixation as given by the proposed algorithm is shown right below the edge map with the fixation.

4 Relationship between Fixation and Segmentation

When the fixation point lies inside a homogeneous region with no strong internal textures, the exact location of the fixation with respect to the region boundary does not affect the segmentation result. It is the same closed contour for any fixation point inside the region. However, there are scenarios when change in fixation inside the region changes the segmentation output. It happens generally when only static monocular cues (color and texture) are used to generate the probabilistic boundary edge map as it leaves strong internal edges in the edge map. There are essentially three such scenarios: 1) when smaller regions are fully contained in the original region (or object); 2) in the presence of dominant internal textures and complex lighting effects; 3) when the fixated region (or object) are extremely concave and has long and thin structures.

4.1 Case 1: Closed Regions Inside an Object

Such objects (e.g., a face) have smaller objects (e.g., eyes and mouth) contained fully inside of them. Given the probabilistic boundary edge map (see Fig. 8), fixations on the smaller regions (or objects) result in the segmentation of those regions as shown in Fig. 8. It is intuitive to see

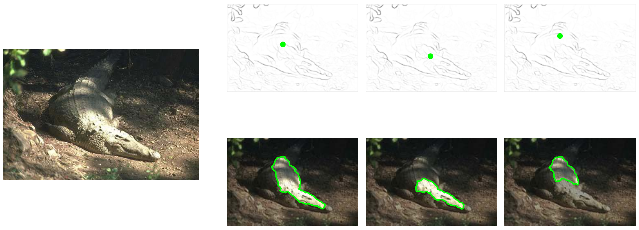


Fig. 9. The densely textured crocodile image is shown on the left. The top row of images contains fixations at different locations on the crocodile overlaid on its inverse probabilistic boundary edge map, while the bottom row of images contains the corresponding segmentation obtained using the proposed algorithm.

that fixating on eyes and mouth should make the visual system see those parts of the face, whereas fixation anywhere else on the face should make the entire face more perceptible. So, such variation in the segmentation with the changing fixation locations is desirable, and makes the proposed algorithm closer to how the human visual system might look at objects like faces. If, however, stereo or motion cues were used and there is no nonrigid motion of the facial features, the internal edges on the face corresponding to the eyes and the lips would vanish and all fixations on the face would result in the same segmentation, the entire face.

But a probabilistic boundary edge map with strong and valid internal edges can be generated even in the presence of motion or stereo cues. For instance, consider that the person whose face we considered above is laughing even as he moves his face. In that case, the edges along the mouth have different flow across them, making them strong boundary edges. The final probabilistic edge map will have strong internal edges corresponding to the boundaries of the mouth and obviously the boundary contour of the face. (Such a probabilistic edge map would be akin to the one with the static monocular cues only.) Now, once again, fixating on the mouth will segment that mouth, whereas fixating anywhere else on the face outside of the mouth will give us the entire face, similar to what happened in the face example stated above. In these circumstances, not getting the same closed contour for all the fixation points inside of a contour is justified.

4.2 Case 2: Texture and Complex Lighting Effects

This case arises when we process single image only, meaning that there is no binocular or motion cues to remove the internal edges from the edge map. Although Malik et al. [25] can handle homogeneous textures using textons, nonhomogeneous textures are hard to tackle and it creates spurious internal edges and disappearance of some boundary edges. Another factor contributing significant spurious internal edges is complex lighting effects on the object. See Fig. 9, an image of a crocodile in the wild. Its probabilistic boundary edge map clearly shows how these two factors have given rise to spurious internal and weak boundary edges, causing significant variation in the segmentation as the fixation shifts from one location to another on the body of the crocodile. Such variation in segmentation with fixation is not desirable, but it can only be fixed either using binocular and (or) motion cues as explained in Section 3.2 or high-level information shape information such as knowledge of what a crocodile looks like and how it can deform its body.

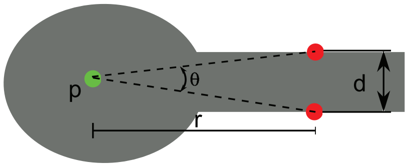


Fig. 10. The problem of thin elongated structure along the radial axis. P is the pole (or fixation) inside the structure with an elongated part of constant width d. θ is the angle any two opposite points along the two parallel sides of the elongated structure at a distance r away from the pole. The parallel lines appear merged to the Point P if $\theta < 1^\circ$ for the farthest point along the parallel sides of the structure.

4.3 Case 3: Concave Shapes with Thin Structures

The location of fixation inside a concave region with thin elongated structures can affect the segmentation output as the thin structures get merged in the polar space due to fixed sampling along the angular axis. While converting the probabilistic boundary edge map from the Cartesian to the polar space is an important step of the proposed segmentation algorithm (Section 3.3), it also causes a slight problem for shapes with thin structures and when the fixation lies sufficiently far away from these thin structures.

Let us understand why having a thin structure can change segmentation output with changes in the fixation location. Referring to Fig. 10, for the elongated part of the shape, the pair of points separated by a distance d and at a distance r away from the pole subtends an angle of θ (in radian) at the pole P such that $\theta \approx \frac{d}{r}$. If we choose the granularity to be 1 degree along the angular axis, the subtended angle θ should be greater than $\frac{\pi}{180}$ for the farthest point on the thin structure of any shape. In other words, for a thin structure of constant thickness d, the farthest point on the structure should be at most at a distance r away from the pole to stay separated in its polar image where $r < \frac{d*180}{\pi}$.

Thin elongated structure that does not satisfy the condition stated above merges to form a line and hence the proposed segmentation method is unable to trace the boundary of the thin structure exactly. See how the fixation on the neck of the Giraffe in Fig. 11a results in the partial detection of the rear leg as the optimal path through the polar edge map cuts in the middle of that leg (Figs. 11b and 11d). Look at the blown-up image of the portion in the polar space where the path cuts the leg prematurely (Fig. 11c) and

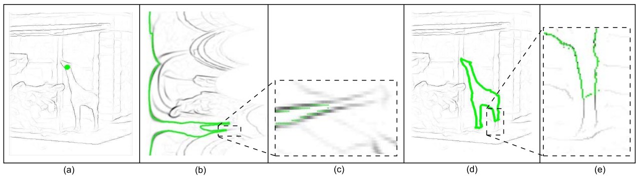


Fig. 11. The problem of merging in the presence of a thin elongated structure. (a) The inverse probabilistic boundary edge map of an image containing a Giraffe with the fixation shown by the green dot. (b) The optimal path through the polar transformation of the edge map. (c) The part of the leg merged together in the polar space is highlighted. (d) The optimal polar path in the Cartesian space. (e) The highlighted portion of the leg in the Cartesian space.

thus an edge is hallucinated in the Cartesian space (Fig. 11e). However, if the fixation is made close to the leg in the Giraffe in Fig. 11, the exact contour of the leg will be revealed fully. Keeping that in mind, we propose a multiple fixation strategy to obtain the boundary of such shapes exactly.

5 Multiple Fixation-Based Segmentation

So far, we have described segmentation for a given fixation. Our objective now is to refine that segmentation by making additional fixations inside the initial segmentation to reveal any thin structures not found in the initial segmentation. Detecting these thin structures can be expensive and complicated if we choose to fixate at every location inside the region. We are going to instead fixate at only a few "salient" locations and incrementally refine the initial segmentations as the new details are revealed. This way we can be certain of not missing any complicated parts of the shape. But where are these salient locations?

5.1 Locations to Make Additional Fixations

The "salient" locations inside the segmentation correspond to those significant changes in the region boundary that results in the protrusion of the contour away from the center. Although there can be many ways to identify these locations, the simplest and fastest way to find them is through the skeleton of the segmented region. It represents the basic shape of the region boundary. We select the junctions of the skeleton as the salient locations as a junction is guaranteed to be present if the boundary has any protruding part.

Although skeleton extraction based on thinning is generally sensitive to slight variations in the region boundary, the divergence-based skeleton extraction proposed by Dimitrov et al. [12] is stable and does not lead to spurious junctions. In fact, using the threshold on the divergence value (which is 0.4 for all our experiments), the spurious junctions arisen due to slight change along the boundary contour can be completely avoided. Besides, the purpose of extracting the skeleton is only to select other possible fixation points inside the segmented region and not to use it to refine the segmentation per se. Thus, the exact topology of the skeleton does not matter to the task at hand. More importantly, choosing fixation points on the skeleton meets the single most important criterion for our segmentation refinement algorithm to succeed: *The fixation points must lie inside the segmented region.*

From the set of junctions in the skeleton, we choose the junction closest to the current fixation point. For example, in Fig. 12, the blue dot in (e) is the next fixation point selected by our algorithm because it is the closest junction on the skeleton (d) of the current segmentation in (c) to the current fixation point (the green dot) in (b). To avoid fixating at the same location twice during the refinement process, all the previous fixations are stored and are used to verify whether the new junction has been fixated previously as all of the junctions are fixated serially. Also, after making a series of fixations, the closest junction is found as the one at the minimum distance from any element in the set of already fixated locations.

5.2 Refining Initial Segmentation

Now, the question is how do we refine the initial segmentation by incorporating new details revealed by making additional fixations? There are two aspects of this process that we should emphasize at the outset: First, the fixations are made in a sequence and, in every step of the process, the boundary edge map is updated to carry the information about the part of region contours found by the valid previous fixations; second, only if the new fixation traces all the known region contours from previous steps, the additional contours revealed by the new fixation are incorporated to refine the segmentation further.

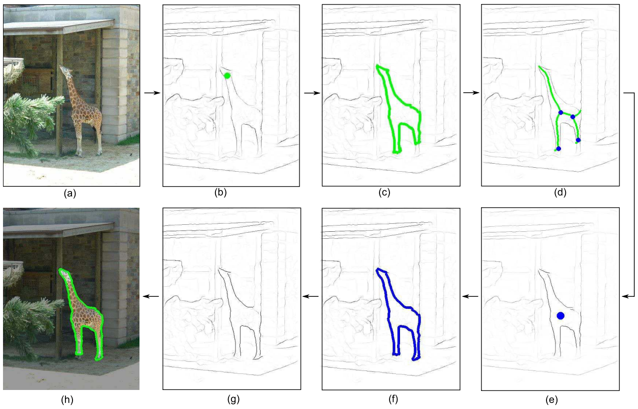


Fig. 12. Multiple fixations to refine the initial segmentation. (a) The original color image containing the Giraffe (the object of interest). (b) The inverse probabilistic boundary edge map and the given fixation (the green circular dot). (c) The segmentation result for the given fixation. (d) The skeleton of the current segmentation with detected junctions shown by the blue circular dots. The junctions not marked are too close to the original fixation. (e) The next fixation (the blue circular dot) in the modified edge map. (f) The segmentation for the new fixation. (g) The modified edge map after incorporating additional information revealed by the new fixation. (h) The final segmentation after fixating at all the other junctions.

At every stage of the refinement process, there is a segmentation mask of the fixated region. The edge fragments that lie along the region boundary and are sufficiently long ($\geq$10 pixels in our experiments) are considered the correct region contours. Accordingly, the probabilistic boundary edge map (in the Cartesian space) is modified such that all the edge pixels along these contours are assigned a probability of 1.0. For any additional fixation, the modified edge map is used to find the corresponding segmentation.

If the segmentation for a new fixation does not trace almost all the known contour pixels, the corresponding segmentation is not considered valid for refining the current segmentation. However, if the new segmentation traces most of the known contours, say, 95 percent (for our experiments) of all the known edge pixels along the contour, the new segmentation is combined with the current segmentation in a binary OR manner. Using the updated current segmentation, the probabilistic boundary edge map is modified to include any new contours revealed by this fixation. The process of refinement stops when all the salient locations have been fixated. Fig. 12e shows the probabilistic boundary edge map refined using the previous segmentation shown in Fig. 12c. Additionally, Fig. 13 shows how the examples of refined boundary edge maps in the third column, and also shows the multiple fixation refinement process successfully reveals the thin structures of the objects.

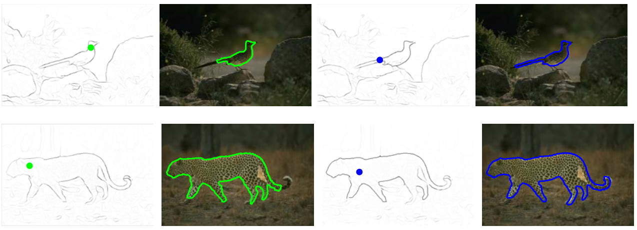


Fig. 13. Segmentation refinement using multifixation segmentation strategy. Column 1: The inverse probabilistic boundary edge map with the first fixation. Column 2: Segmentation result. Column 3: The modified edge map with the next most important fixation. Column 4: Segmentation result for the next fixation.

6 Experiments and Results

6.1 Segmentation Accuracy

Our data set is a collection of 20 videos with average length of seven frames and 50 stereo pairs along with their ground-truth segmentation. For each sequence and stereo pair, only the most prominent object of interest is identified and segmented manually to create the ground-truth foreground and background masks. The fixation is chosen randomly anywhere on this object of interest. The videos used for the experiment are diverse: stationary scenes captured with a moving camera, dynamic scenes captured with a moving camera, and dynamic scenes captured with a stationary camera.

The segmentation output of our algorithm is compared with the ground-truth segmentation in terms of the F-measure defined as $2PR/(P+R)$, where P stands for the precision which calculates the fraction of our segmentation overlapping with the ground truth and R stands for recall which measure the fraction of the ground-truth segmentation overlapping with our segmentation.

Table 1 shows that after adding motion or stereo cues to the color and texture cues the performance of the proposed method improves significantly. With color and texture cues only, the strong internal edges prevent the method from tracing the actual depth boundary. (See Fig. 15, Row 2.) However, the motion or stereo cues clean the internal edges as described in Section 3 and the proposed method finds the correct segmentation (Fig. 15, Row 3).

To also evaluate the performance of the proposed algorithm in the presence of the color and texture cues only, the images from the Alpert image database [2] have been used. The Berkeley edge detector [25] provides the probabilistic boundary maps of these images. The fixation on the image is chosen at the center of the bounding box around the foreground. In case of multiple objects, a fixation point is selected for each of them. For a fixation point, our algorithm finds the region enclosed by the depth boundary in the scene, where it is difficult to find only the color and texture cues. However, when the color and texture gradient is such that it is higher at the pixels on depth boundary than that inside the object, the segmentation results are consistent with the expected outcome. As we can see in Table 2, we perform better than even the state of the art for the set of images with two objects and close to [2], [4] for the images with a single object. The consistent performance of our algorithm for two types of images in the data set can be attributed to the scale-invariance property of our algorithm. Also, the definition of segmentation in [4] is such that, for a selected seed on any of the two horses in Fig. 14, left, both horses will be segmented. This illustrates that seed point in [4] has no significance other than selecting a good initial segment to start the processing of segmentation. In contrast, our segmentation finds only the horse being fixated, making the so-called "seed point" of our algorithm a meaningful input which identifies the object of interest.

TABLE 1
The Performance of Our Segmentation for the Videos and the Stereo Pairs

For videos	F-measure
With Motion	0.97 ± 0.02
Without Motion	0.62 ± 0.02
For stereo pairs	
With Stereo	0.95 ± 0.01
Without Stereo	0.65 ± 0.02

See Fig. 15.

TABLE 2
One Single Segment Coverage Results

Algorithm	Score (Single)	Score (Double)
Our Method	0.83 ± 0.019	0.81 ± 0.03
Bagon et al. [4]	0.87 ± 0.010	N.A.
Alpert et al. [2]	0.86 ± 0.012	0.68 ± 0.05
NCut [35]	0.72 ± 0.012	0.58 ± 0.06
MeanShift [43]	0.57 ± 0.023	0.61 ± 0.02

The scores (F-measure) for other methods except [4] are taken from the website hosting the database. N.A. means the score not available for

In Fig. 16, we provide a visual comparison between the output of the proposed segmentation and the interactive GrabCut algorithm [33] and Normalized Cut [37] for some of the difficult images from the Berkeley Segmentation Database [25]. For normalized cut, the best parameter (between 5 and 20) for each image is manually selected and the corresponding segmentation is shown in the last row of Fig. 16.

6.2 Semantic Significance: An Empirical Study

In the experiments so far, we have found that the proposed method segments the fixated region (or object) accurately and consistently, especially in the presence of both binocular and statoc cues. But in the case of static cues only, a fixation on an object often results in a segmentation that is mostly just a part of that object in the scene. What is interesting, however, is to study if there is a consistency in segmenting that part if we fixate at the same location inside an object as it appears in different images. In other words, we empirically study how semantically meaningful are the regions segmented by the proposed algorithm so that the

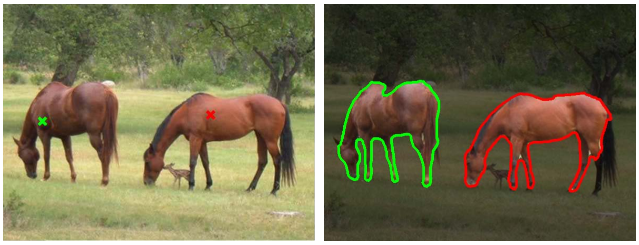


Fig. 14. Left: An image with two fixations (the symbol "X"). Right: The corresponding segmentation for these fixations as given by the proposed framework.

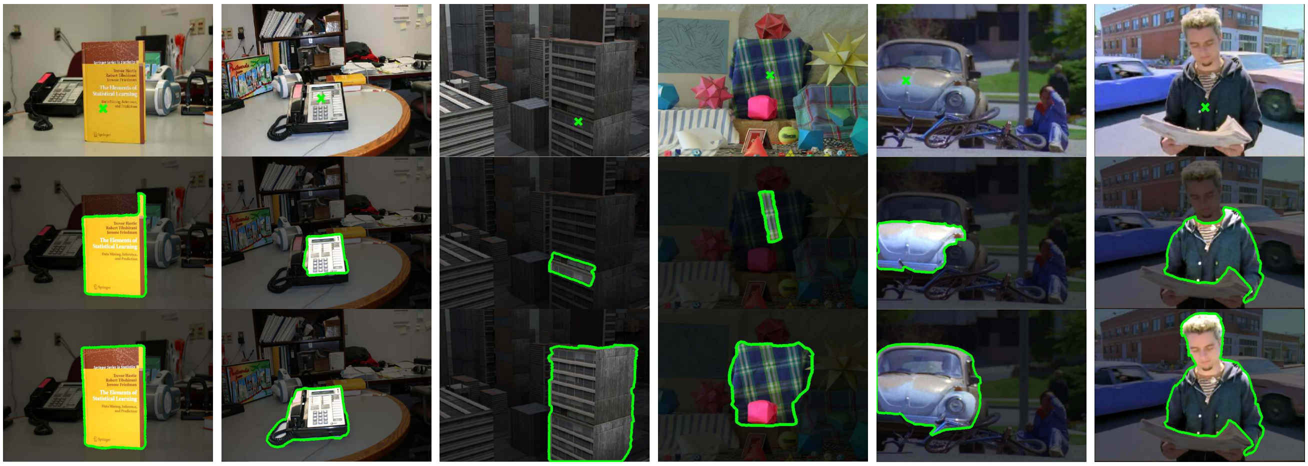


Fig. 15. Columns 1-3: A moving camera and stationary objects. Column 4: An image from a stereo pair. Column 5: A moving object (car) and a stationary camera. Column 5: Moving objects (human, cars) and a moving camera. Row 1: The original images with fixations (the green “X”). Row 2: The segmentation results for the fixation using statoc cues only. Row 3: The segmentation for the same fixation after combining motion or stereo cues with monocular cues.

algorithm can be used as a useful first step in the object recognition process.

For this study, we are going to use the ETHZShape database [15], which contains 255 images of five different objects, namely Giraffes, Swans, Bottles, Mugs, and Apple-logos. As the final probabilistic boundary edge detector is calculated using static cues only, the fixation location plays an important role in deciding what we get as the segmentation output. For instance, fixation on the neck of a Giraffe results in the segmentation of its neck, see Fig. 17. If all the internal texture edges, however, are suppressed using, say, binocular cues, fixating anywhere on the Giraffe would lead to the segmentation of the entire Giraffe. Thus, it is important to choose the same fixation location inside the object, so that the variation due to this change in fixation location can be discounted for.

We need to make sure that we fixate at all the different parts of an object. We avoid selecting these fixations manually as our selection would be heavily biased by the individual preference. Instead, we use the shape of the object to find the salient locations inside it to fixate and the segmented regions for these fixations are then manually labeled as a part if it appears so. This way, the parts are created from the low-level information and are only labeled by human subjects.

The question now is what are those salient locations to fixate at and will those fixations be at similar locations inside the object across different instances of that object in the database? We hand segment the object in each image (we randomly select one in the image with multiple objects), and fixate at the middle of the branches of the skeleton of the binary object mask. A branch of the skeleton correspond

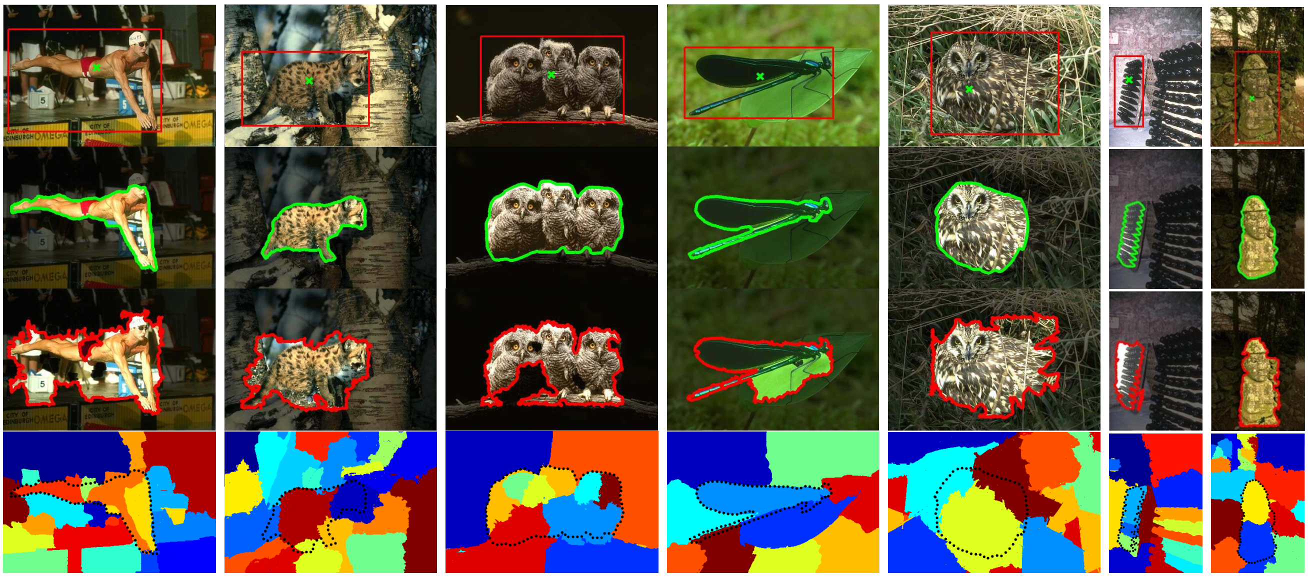


Fig. 16. The first row contains images with the fixation shown by the green X. Our segmentation for these fixations is shown in the second row. The red rectangle around the object in the first row is the user input for the GrabCut algorithm [34]. The segmentation output of the iterative GrabCut algorithm (implementation provided by www.cs.cmu.edu/~mohitg/segmentation.htm) is shown in the third row. The last row contains the output of the Normalized cut algorithm with the region boundary of our segmentation overlaid on it.

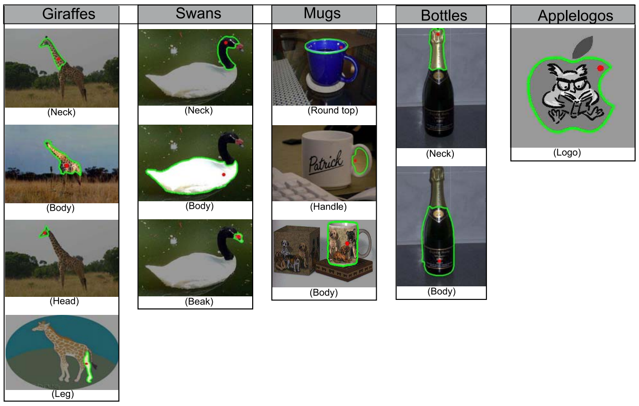


Fig. 17. Examples of segmented object parts. The red circular dot shows the fixation point and the green contour is the boundary of the segmented region for that fixation point. Giraffes, Swans, Mugs, Bottles, and Applelogos are found to have four, three, three, two, and one part (s), respectively.

to an object part such as neck, leg, etc. The junctions in the skeleton correspond to where the parts combine to form the complete object.

We fixate at all the salient locations on the objects and collect the segmented regions corresponding to those fixations. Then, we examine every segmentation manually and label the segmented region as an object part if it results from fixations at similar location on the object in most images in the database. See Fig. 17 for a sample of the parts of all five objects. Obviously, the number of parts for an object depends upon the complexity of its shape. The Giraffe has the highest number of parts whereas Applelogo has the least. (For Applelogos, we don't include the leaf of the apple as its part as, in our work, the object is a compact region.)

For each object, we count the number of times an object part is fixated and what percentage of the total number of fixations resulted in the segmentation of that part as well as that of the entire object and semantically meaningless parts. These numbers are shown in the corresponding row of the table for that object. See Table 3. Different parts have different likelihood of being segmented on being fixated. But some parts like the handle of a Mug, the entire Applelogo, the neck of a Swan, etc., have high likelihood of being segmented on being fixated.

Another important statistic of interest is how often one of the fixations on the object results in the segmentation of the entire object. For that, we calculate the overlap of the biggest segmented region for each object with the hand-segmented object mask. We calculate the mean of the overlap of the biggest region over all images in the database. See Table 4. The likelihood of segmenting an entire object is dependent upon how textured the object is. The Applelogos are segmented entirely by a single fixation, whereas bottles mostly have labels on them and are generally only segmented into its upper or lower half.

7 Fixation Strategy

The proposed segmentation method clearly depends on the fixation point, and thus it is important to select the fixations automatically. Fixation selection is a mechanism that depends on the underlying task as well as other senses (like sound). In the absence of such information, one has to concentrate on generic visual solutions. There is a significant amount of research done on the topic of visual attention [30], [41], [36], [45], [20], [34], [9] primarily to find the salient locations in the scene where the human eye may fixate. For our segmentation framework, as the next section shows, the fixation just needs to be inside the objects in the scene. As long as this is true, the correct segmentation will be obtained. Fixation points can be found using low-level features in the scene and, in that respect, the recent literature on features comes in handy [25], [28]. Although we do not yet have a definite way to automatically select fixations, we can easily generate potential fixations that lie inside most of the objects in a scene.

TABLE 3
Detection of a Fixated Object Part

(a) Swans

	Neck	Body	Beak	Whole	Non-parts
Neck	55.22	0	0.00	26.8	17.91
Body	0.00	28.91	0.00	37.34	33.73
Beak	0.00	0.00	87.5	0.00	12.50

(b) Giraffes

	Head	Neck	Leg	Body	Whole Obj	Non-parts
Head	71.66	0.00	0.00	0.00	16.67	11.67
Neck	0.00	75.86	0.00	0.00	6.87	10.34
Leg	0.00	0.00	35.00	0.00	3.57	61.40
Body	0.00	0.00	0.00	26.92	25.0	48.07

(c) Mugs

	Round top	Handle	Body	whole Obj	Non-parts
Round Top	90.00	0.00	0.00	0.00	10.00
Handle	98.9	0.00	0.00	0.00	1.1
Body	0.00	0.00	14.28	47.40	38.31

(d) Bottles

	Neck	Body	Whole Obj	Non-parts
Neck	44.61	0.00	18.46	36.92
Body	0.00	28.16	16.90	54.92

(e) Applelogos

	Applelogos	Non parts
Body	91.8	8.2

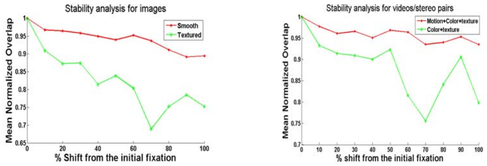
Each entry (i, j) of the table is the percentage of total fixations on the part i that resulted in the segmentation of the part j, which is decided manually.

7.1 Stability Analysis

Here, we verify our claim that the optimal closed boundary for any fixation inside a region remains the same. The possible variation in the segmentation will occur due to the presence of bright internal edges in the probabilistic boundary edge map. To evaluate the stability of segmentation with respect to the location of fixation inside the object, we devise the following procedure: Choose a fixation roughly at the center of the object and calculate the optimal closed boundary enclosing the segmented region. Calculate the average scale S_{avg} of the segmented region as $\sqrt{Area/\pi}$.

TABLE 4
The Mean of the Highest Overlap ($\times 100$) for Each Image

Object name	Mean overlap
Giraffes	71.47
Swans	86.85
Mugs	84.13
Applelogos	95.13
Bottles	52.96



(a) (b)

Fig. 18. Stability curves for region segmentation variation with respect to change in fixation locations.

Now, the new fixation is chosen by moving away from the original fixation in random direction by $n \cdot S_{avg}$, where $n = \{0.1, 0.2, 0.3, \ldots, 1\}$. If the new fixation lies outside the original segmentation, a new direction is chosen for the same radial shift until the new fixation lies inside the original segmentation. The overlap between the segmentation with respect to the new fixation, R_n, and the original segmentation, R_o, is given by $\frac{|R_o \cap R_n|}{|R_o \cup R_n|}$.

We calculated the overlap values for 100 textured regions and 100 smooth regions from the BSD and Alpert Segmentation Database. It is clear from the graph in Fig. 18a that the overlap values are better for the smooth regions than for the textured regions. Textured regions might have strong internal edges, making it possible for the original optimal path to be modified as the fixation moves to a new location. However, for smooth regions, there is a stable optimal path around the fixation; it does not change dramatically as the fixation moves to a new location. We also calculate the overlap values for the 100 frames from video sequences; first with their boundary edge map given by Martin et al. [24] and then using the enhanced boundary edge map after combining motion cues. The results are shown in Fig. 18b. We can see that the segmentation becomes stable as motion cues suppress the internal edges and reinforce the boundary edge pixels in the boundary edge map [24].

8 Discussion and Future Work

The proposed framework has successfully separated the segmentation process into cue processing and segmenting the region containing a given fixation point. The visual cues are used only to influence the probability of the pixels in the image to be the depth/object boundary. After calculating the probabilistic boundary edge map, the segmentation of the fixated object/region becomes a well-defined binary labeling problem in the polar space.

An important advantage of separating cue processing from segmentation step is that these two steps form a feedback loop between them. The forward process of generating a closed contour given a point inside the probabilistic boundary edge map is a bottom-up step, whereas using the resulting region to either modify the probabilistic edge map, say, using shape information, or to select the next fixation point using that information stored in the region is a top-down process. The multiple fixation based refinement of initial segmentation described in Section 5 is an example of an interaction between the bottom-up and the top-down process. In this case, the top-down process was using only the shape of the segmented region to predict the next location to fixate to refine the previous segmentation.

The top-down process can be more elaborate. In addition to using the part of an object segmented using the first fixation point in Fig. 17 to predict the fixation point inside the other part of that object, the shape of that part can modify the probabilistic boundary map such that the edge pixels along the expected contour is strengthened. A similar strategy to combine the top-down with bottom-up process has been employed in [13], wherein the authors first focus on a component of a face and use the prior knowledge about the shape of that component to segment it better.

9 Conclusion

We proposed here a novel formulation of segmentation in conjunction with fixation. The framework combines static cues with motion and/or stereo to disambiguate between the internal and the boundary edges. The approach is motivated by biological vision, and it may have connections to neural models developed for the problem of border ownership in segmentation [11]. Although the framework was developed for an active observer, it applies to image databases as well, where the notion of fixation amounts to selecting an image point which becomes the center of the polar transformation. Our contribution here was to formulate an old problem—segmentation—in a different way and show that existing computational mechanisms in the state-of-the-art computer vision are sufficient to lead us to promising automatic solutions. Our approach can be complemented in a variety of ways, for example, by introducing a multitude of cues. An interesting avenue has to do with learning models of the world. For example, if we had a model of a "horse," we could segment the horses more correctly in Fig. 14. This interaction between low-level bottom-up processing and high-level top-down attentional processing, is a fruitful research direction.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-Tuned Salient Region Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.

[2] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2007.

[3] P. Arbelaez and L. Cohen, "Constrained Image Segmentation from Hierarchical Boundaries," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 454-467, 2008.

[4] S. Bagon, O. Boiman, and M. Irani, "What Is a Good Image Segment? A Unified Approach to Segment Extraction," *Proc. 10th European Conf. Computer Vision*, pp. 30-44, 2008.

[5] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive Image Segmentation Using an Adaptive GMMRF Model," *Proc. European Conf. Computer Vision*, pp. 428-441, 2004.

[6] Y.Y. Boykov and M.P. Jolly, "Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in n-d Images," *Proc. Eighth IEEE Int'l Conf. Computer Vision*, pp. 105-112, 2001.

[7] Y.Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124-1137, Sept. 2004.

[8] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High Accuracy Optical Flow Estimation Based on a Theory for Warping*, pp. 25-36. Springer, 2004.

[9] N.D.B. Bruce and J.K. Tsotsos, "Saliency, Attention, and Visual Search: An Information Theoretic Approach," *J. Vision,* vol. 9, no. 3, pp. 1-24, 2009.

[10] M. Cerf, J. Harel, W. Einhäuser, and C. Koch, "Predicting Human Gaze Using Low-Level Saliency Combined with Face Detection," *Proc. Neural Information Processing Systems,* 2008.

[11] E. Craft, H. Schütze, E. Niebur, and R. von der Heydt, "A Neural Model of Figure-Ground Organization," *J. Neurophysiology,* vol. 6, no. 97, pp. 4310-4326, 2007.

[12] P. Dimitrov, C. Phillips, and K. Siddiqi, "Robust and Efficient Skeletal Graphs," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 417-423, 2000.

[13] L. Ding and M.A. Martinez, "Features versus Context: An Approach for Precise and Detailed Detection and Delineation of Faces and Facial Features," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 11, pp. 2022-2038, Nov. 2010.

[14] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *Int'l J. Computer Vision,* vol. 59, no. 2, pp. 167-181, 2004.

[15] V. Ferrari, T. Tuytelaars, and L.V. Gool, "Object Detection by Contour Segment Networks," *Proc. European Conf. Computer Vision,* pp. 14-28, June 2006.

[16] M. Gur, A. Beylin, and D.M. Snodderly, "Response Variability of Neurons in Primary Visual Cortex (V1) of Alert Monkeys," *J. Neuroscience,* vol. 17, pp. 2914-2920, 1997.

[17] J.M. Henderson and A. Hollingworth, *Eye Movements during Scene Viewing: An Overview.* Oxford, 1998.

[18] J.M. Henderson, C.C. Williams, M.S. Castelhano, and R.J. Falk, "Eye Movements and Picture Processing during Recognition," *Perception and Psychophysics,* vol. 65, pp. 725-734, 2003.

[19] A. Hollingworth, G. Schrock, and J.M. Henderson, "Change Detection in the Flicker Paradigm: The Role of Fixation Position within the Scene," *Memory and Cognition,* vol. 29, pp. 296-304, 2001.

[20] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1254-1259, Nov. 1998.

[21] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother, "Bi-Layer Segmentation of Binocular Stereo Video," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 407-414, 2005.

[22] E. Kowler and R.M. Steinman, "Small Saccades Serve No Useful Purpose: Reply to a Letter by R. W. Ditchburn," *Vision Research,* vol. 20, pp. 273-276, 1980.

[23] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision* , vol. 60, no. 2, pp. 91-110, 2004.

[24] D. Martin, C. Fowlkes, and J. Malik, "Learning to Detect Natural Image Boundaries Using Local Brightness, Color and Texture Cues," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 5, pp. 530-549, May 2004.

[25] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics," *Proc. Eighth IEEE Int'l Conf. Computer Vision,* vol. 2, pp. 416-423, July 2001.

[26] A.M. Martínez, P. Mittrapiyanuruk, and A.C. Kak, "On Combining Graph-Partitioning with Non-Parametric Clustering for Image Segmentation," *Computer Vision and Image Understanding,* vol. 95, pp. 72-85, July 2004.

[27] S. Martinez-Conde, S.L. Macknik, and D.H. Hubel, "The Role of Fixational Eye Movements in Visual Perception," *Nature Rev. Neuroscience,* vol. 5, pp. 229-240, 2004.

[28] K. Mikolajczyk and C. Schmid, "An Affine Invariant Interest Point Detector," *Proc. Seventh European Conf. Computer Vision,* pp. 128-142, 2002.

[29] A.S. Ogale and Y. Aloimonos, "A Roadmap to the Integration of Early Visual Modules," *Int'l J. Computer Vision,* vol. 72, no. 1, pp. 9-25, Apr. 2007.

[30] D. Parkhurst, K. Law, and E. Niebur, "Modeling the Role of Salience in the Allocation of Overt Visual Attention," *Vision Research,* vol. 42, pp. 107-23, 2000.

[31] U. Rajashekar, I. van der Linde, A.C. Bovik, and L.K. Cormack, "Gaffe: A Gaze-Attentive Fixation Finding Engine," *IEEE Trans. Image Processing,* vol. 17, no. 4, pp. 564-573, Apr. 2008.

[32] A.L. Rothenstein and J.K. Tsotsos, "Attention Links Sensing to Recognition," *Image Vision Computing,* vol. 26, no. 1, pp. 114-126, 2008.

[33] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': Interactive Foreground Extraction Using Iterated Graph Cuts," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 309-314, 2004.

[34] J.T. Serences and S. Yantis, "Selective Visual Attention and Perceptual Coherence," *Trends in Cognitive Sciences,* vol. 10, no. 1, pp. 38-45, 2006.

[35] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.

[36] C. Siagian and L. Itti, "Rapid Biologically-Inspired Scene Classification Using Features Shared with Visual Attention," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 2, pp. 300-312, Feb. 2007.

[37] A.K. Sinop and L. Grady, "A Seeded Image Segmentation Framework Unifying Graph Cuts and Random Walker Which Yields a New Algorithm," *Proc. IEEE 11th Int'l Conf. Computer Vision,* pp. 1-8, 2007.

[38] R.H. Steinberg, M. Reid, and P.L. Lacy, "The Distribution of Rods and Cones in the Retina of the Cat (Fells Domestica)," *J. Computational Neuroscience,* vol. 148, pp. 229-248, 1973.

[39] X.Y. Stella and J. Shi, "Grouping with Bias," *Proc. Neural Information Processing Systems,* 2001.

[40] O.M. Thomas, B.G. Cumming, and A.J. Parker, "A Specialization for Relative Disparity in V2," *Nature Neuroscience,* vol. 5, no. 5, pp. 472-478, May 2002.

[41] A. Torralba, A. Oliva, M.S. Castelhano, and J.M. Henderson, "Contextual Guidance of Eye Movements and Attention in Real-World Scenes: The Role of Global Features in Object Search," *Psychological Rev.*, vol. 113, no. 4, pp. 766-786, 2006.

[42] A. Toshev, A. Makadia, and K. Daniilidis, "Shape-Based Object Recognition in Videos Using 3D Synthetic Object Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[43] Z.W. Tu and S.C. Zhu, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 603-619, May 2002.

[44] O. Veksler, "Star Shape Prior for Graph-Cut Image Segmentation," *Proc. 10th European Conf. Computer Vision,* vol. 3, pp. 454-467, 2008.

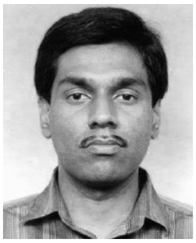
[45] D. Walther and C. Koch, "Modeling Attention to Salient Proto-Objects," *Neural Networks,* vol. 19, no. 4, pp. 1395-1407, Apr. 2006.

[46] K.C. Winkler, R.W. Williams, and P. Rakic, "Photoreceptor Mosaic: Number and Distribution of Rods and Cones in the Rhesus Monkey," *J. Computational Neuroscience,* vol. 297, pp. 499-508, 1990.

Ajay K. Mishra received the BTech degree from the Indian Institute of Technology, Kanpur and the PhD Degree from the National University of Singapore in 2003 and 2011, respectively. Currently, he is a research associate at the University of Maryland. From 2003 to 2005, he was a design engineer in STMicroelectronics Pvt Ltd. He won the first prize in the Semantic Robot Vision Challenge 2008. His research interests are in the development of fixation-based vision solutions for robotics and multimedia systems.

Yiannis Aloimonos received the Diplome in mathematics in Greece in 1982 and the PhD degree in computer science in Rochester, New York, 1987. He is a professor of computational vision in the Department of Computer Science at the University of Maryland, College Park, and the director of the Computer Vision Laboratory. His research interests are in the integration of vision, action, and cognition.

Loong-Fah Cheong received the BEng degree from the National University of Singapore and the PhD degree from the University of Maryland at College Park in 1990 and 1996, respectively. In 1996, he joined the Department of Electrical and Computer Engineering, National University of Singapore, where he is now an associate professor. His research interests include 3D motion perception, 3D navigation, and multimedia system analysis.



Ashraf A. Kassim received the BEng (First Class Honors) in electrical engineering from the National University of Singapore (NUS) and had worked on machine vision systems at Texas Instruments before proceeding to receive the PhD degree in electrical and computer engineering from Carnegie Mellon University in 1993. He has been with the Electrical and Computer Engineering Department at the National University of Singapore since 1993 and is currently a vice dean of the engineering school. His research interests include image analysis, machine vision, video/image processing, and compression. He is a member of the IEEE.